ELSEVIER

Theoretical Computer Science 143 (1995) 113–122

Theoretical Computer Science

Note

Shortest consistent superstrings computable in polynomial time*

Tao Jiang*, Vadim G. Timkovsky

Department of Computer Science and Systems, McMaster University, Hamilton, Ont., Canada L8S 4K1

Received July 1994; revised September 1994
Communicated by M. Nivat

Abstract

The shortest consistent superstring problem is, given a set of positive strings and a set of negative strings, finding a shortest string including every positive string and no negative string as a substring. This problem is NP-hard and arises in DNA sequencing by hybridization. It is also an extension of the well-known shortest common superstring problem which corresponds to the case when the set of negative strings is empty. In this paper we show that a shortest consistent superstring can be found in polynomial time if (i) a longest common nonsuperstring for the set of negative strings exists or (ii) the number of positive strings is bounded and every symbol of the alphabet appears at the end of some negative string. In the case (i) a longest consistent superstring can also be found in polynomial time.

1. Introduction

Jiang and Li [5, 6] were perhaps the first to pose the *shortest consistent superstring* problem: Given a set P of *positive* strings and a set N of *negative* strings over an alphabet Σ, find a shortest *consistent* (*with* N) *superstring* for P, i.e. a shortest string over Σ including every positive string and no negative string as a substring. Since N will remain invariable further, we will omit the expression in the brackets. Obviously, this problem has a solution only if $P \cap N = \emptyset$. Further we assume (restrictively) that $P \cup N$ is *inclusion free*, i.e., no string in the set includes another as a substring. In particular, this union does not include the empty string. Besides, without loss of generality we assume that there is no one-symbol negative string, say, a because, due

* Supported in part by NSERC Research Grant OGP0046613.
* Corresponding author. Email addresses: {jiang, timko}@macs.mcmaster.ca.

Elsevier Science B.V.
SSDI 0304-3975(94)00252-5

to the inclusion free requirement, no string in $P \cup N$ except a, therefore, no string sought for, contains a. So, a can be deleted from Σ and N.

This problem arises in DNA sequencing by hybridization [1, 11] and the PAC learning model [16] for DNA sequencing [6, 10] and can be considered as a natural extension of the well-known *shortest common superstring* problem [2, 9, 14] which corresponds to the case when $N = \emptyset$. Since the latter is NP-hard, the former is also so and remains NP-hard even if $N \neq \emptyset$ [7]. In the case $P = \emptyset$, the shortest consistent superstring problem becomes trivial. However, replacing the "shortest"requirement by the "longest" one transforms this case into the *longest common nonsuperstring* problem [15] solvable in polynomial time [12, 13]. Consistent superstrings can be considered as intermediate structures between common superstrings and common nonsuperstrings. It prompts the conjecture that there are polynomially solvable nontrivial cases of the shortest consistent superstring problem that somehow are related to longest common nonsuperstrings. As it turns out, these cases exist indeed and arise from the graph representation of common nonsuperstrings that has been proposed in [13].

In this paper we show that a shortest consistent superstring can be found in polynomial time if (i) a longest common nonsuperstring for N exists or (ii) $|P|$ is bounded and every symbol of Σ appears at the end of some string in N. As we will see further, in the case (i) a *longest consistent superstring* can also be found in polynomial time.

We hope that these results will find applications in creating more effective methods for DNA sequencing by hybridization. Besides, the case (i) provides an example of "downfall" phenomenon – an NP-hard problem has a polynomially solvable extension, which is of some theoretical interest.

2. Preliminaries

2.1. Graph representation of common nonsuperstrings

Here we cite some notation and results in [13]. Let α be a string over an alphabet Σ and n be natural. Then α^n denotes the concatenation of n copies of α. The following condition is necessary for the existence of a longest common nonsuperstring for N.

$$\forall a \in \Sigma \ \exists n \geqslant 2: \quad a^n \in N \quad \text{(a quadratical closure of } N\text{)}.$$

Indeed, let this condition be false. Then either $a \in N$ for some $a \in \Sigma$ or $a^n \notin N$ for all natural n. The former is not the case because of the original assumption, the latter implies that a^n is a common nonsuperstring for N and so there is no longest one.

Let $|\alpha|$ be the length of α, ε be the empty string, $|\varepsilon| = 0$, $Pref_k \alpha$ and $Suff_k \alpha$ be the prefix and the suffix of length k of α, where $k = 0, 1, \ldots, |\alpha|$ and $Pref_0 \alpha = Suff_0 \alpha = \varepsilon$. We will write $Pref \alpha$ and $Suff \alpha$ if the length of prefixes and suffixes is immaterial.

The quadratical closure condition implies the following weaker one.

$$\forall a \in \Sigma\ \exists \alpha \in N:\quad a = Suff_1 \alpha \quad \text{(a final closure of } N\text{)}.$$

So, this condition is also necessary for the existence of a longest common nonsuperstring. In the following, we assume that N is finally closed.

For a set V of strings over Σ without ε define a directed graph G_V with vertex set V and arc set E determined by the rule:

$$(\alpha, \beta) \in E \Leftrightarrow Suff_{|\alpha|-1}\alpha = Pref_{|\alpha|-1}\beta.$$

The arc (α, β) arises when $|\alpha| \leqslant |\beta| - 1$ and there is a string of length $|\beta| + 1$ over Σ with prefix α and suffix β. This string is denoted as $[\alpha, \beta]$. Note that $[\alpha, \beta]$ is not necessarily in V. In particular, if $|\alpha| = 1$, then there are arcs from α to all other vertices of G_V and $Suff_{|\alpha|-1}\alpha = \beta$ implies $(\alpha, \beta) \in E$.

Example 2.1. $G_{English}$ contains arcs (*word, order*), (*a, part*), (*there, here*), herein [*word, order*] = *worder*, [*a, part*] = *apart*, [*there, here*] = *there*.[1]

Let $\alpha(i)$ be the symbol at the ith *position* of α, i.e. $\alpha = \alpha(1)\alpha(2)\ldots\alpha(|\alpha|)$, and $V^{\supseteq}$ be the set of all common superstrings for V. A *route* in a graph is denoted by the sequence $\mathscr{A} = (\alpha_1, \alpha_2, \ldots, \alpha_{k-1}, \alpha_k)$ of its vertices such that for all $i = 1, 2, \ldots, k-1$ (α_i, α_{i+1}) is an arc of the graph.[2] $|\mathscr{A}|$ will denote the *length of* $\mathscr{A}$, i.e. $|\mathscr{A}| = k - 1$. For every route $\mathscr{A} = (\alpha_1, \alpha_2, \ldots, \alpha_{k-1}, \alpha_k)$ in G_V define the string

$$f(\mathscr{A}) = \alpha_1(1)\alpha_2(1)\ldots\alpha_{k-1}(1)\alpha_k.$$

Obviously, $f(\mathscr{A}) \in \{\alpha_1, \alpha_2, \ldots, \alpha_{k-1}, \alpha_k\}N^{\supseteq}$.

Remark 2.1. Let Σ^n be the set of all strings of length n over Σ. Then G_{Σ^n} is the well-known de Bruijn's graph [3] and f is a one-to-one correspondence between the set of routes of length $k-1$ in G_{Σ^n} and Σ^{n+k-1}.

For a string α with $|\alpha| > 1$ we call $Pref_n\alpha$ and $Suff_n\alpha$ *proper* if $0 < n < |\alpha|$. Let S be the set of proper suffixes of strings in N. For every nonempty string ω over Σ define the route in G_S

$$g(\omega) = (\alpha_1, \alpha_2, \ldots, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{|\omega|-1}, \alpha_{|\omega|}),$$

where α_i is the longest string of S included in ω as a substring starting from the ith position. Since the final closure condition holds, $a \in \Sigma \Rightarrow a \in S$ and so the choice of α_i is always possible. It is important to observe that the inequality $|\alpha_i| \leqslant |\alpha_{i+1}| + 1$ and

[1] Note that *English* is actually not inclusion free. The inclusion freeness assumption is only needed in the next section.

[2] Unlike a path, a route can intersect itself.

the equality $Suff_{|\alpha_i|-1}\alpha_i = Pref_{|\alpha_i|-1}\alpha_{i+1}$ follow from the fact that α_i and α_{i+1} are included in ω as substrings starting from the ith and $(i+1)$th positions, respectively, and the longest length requirement. Thus, the arc (α_i, α_{i+1}) exists in fact, i.e. the definition of the route $g(\omega)$ is correct. It is easy to see that $f(g(\omega)) = \omega$.

Let Γ_S be a subgraph of G_S with vertex set S and arcs (α, β) so that α is the longest prefix of $|\alpha, \beta|$ contained in $N \cup S$, i.e. among suffixes of N there are no prefixes of $[\alpha, \beta]$ longer than α. Note that this definition is independent of the final closure condition.

Example 2.2. If $N = English$ then the arc $(ove, venir)$ of G_S is not in Γ_S, since $ove = Suff_3 love$, $venir = Suff_5 souvenir$, $[ove, venir] = ovenir$, but $oven = (Pref_4 ovenir) \in N$. Neither is the arc (e, t) of G_S in Γ_S since $e = Suff_1 love$, $t = Suff_1 let$, $[e, t] = et = Pref_2 et = (Suff_2 net) \in S$. However the arc (ee, ea) is in Γ_S since $ee = Suff_2 tree$, $ea = Suff_2 tea$, $[ee, ea] = eea$, where $eea \notin N \cup S \ni ee$.

Let $V^{\supseteq}$ be the set of all common nonsuperstrings for V. The following lemma [13] shows that Γ_S is constructed so that there is a correspondence between the set of routes in Γ_S and $N^{\supseteq}$.

Lemma 2.1. (a) *If $\mathscr{A}$ is a route in Γ_S, then $f(\mathscr{A}) \in N^{\supseteq}$*; (b) *if $\omega \in N^{\supseteq}$, then $g(\omega)$ is a route in Γ_S.*

Remark 2.2. Note that the last vertex of the route $g(\omega)$ is a one-symbol suffix. Besides, an arbitrary route $\mathscr{A} = (\alpha_1, \ldots, \alpha_k)$ in Γ_S can be extended to the route

$$\mathscr{A}' = (\alpha_1, \ldots, \alpha_k, Suff_{|\alpha_k|-1}\alpha_k, Suff_{|\alpha_k|-2}\alpha_k, \ldots, Suff_1 \alpha_k).$$

Lemma 2.1 reduces a consideration of common nonsuperstrings for N to the analysis of the graph Γ_S. Bounded length of common nonsuperstrings for N means bounded length of routes in Γ_S, i.e. Γ_S is acyclic. Hence, we have the following theorem [13].

Theorem 2.1. *If the graph Γ_S is acyclic, then $\mathscr{A}$ is the longest path in it if and only if $f(\mathscr{A})$ is a longest common nonsuperstring for N. If $\mathscr{A} = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ is a closed route in Γ_S, i.e. $\alpha_1 = \alpha_k$, then $[Pref_{k-1} f(\mathscr{A})]^n \in N^{\supseteq}$ for any natural n so there is no longest common nonsuperstring for N.*

Thus, a necessary and sufficient condition for the existence of a longest common nonsuperstring for N is the acyclicity of Γ_S.

2.2. Wreaths

Let $\mathscr{A} = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ and $\mathscr{B} = (\beta_1, \beta_2, \ldots, \beta_l)$ be two routes and $k \leqslant l$. If $\alpha_k = \beta_1$, then $\mathscr{A} \circ \mathscr{B}$ will denote the *composition of* $\mathscr{A}$ and $\mathscr{B}$, i.e. $\mathscr{A} \circ \mathscr{B} =$

$(\alpha_1, \alpha_2, \ldots, \alpha_k, \beta_2, \ldots, \beta_l)$. We call $\mathscr{A}$ *a head* of $\mathscr{B}$, if $\alpha_1 = \beta_1, \alpha_2 = \beta_2, \ldots, \alpha_k = \beta_k$, a *tail* of $\mathscr{B}$, if $\alpha_1 = \beta_{l-k+1}$, $\alpha_2 = \beta_{l-k+2}, \ldots, \alpha_k = \beta_l$, a *subroute* of $\mathscr{B}$, if $\alpha_1 = \beta_i$, $\alpha_2 = \beta_{i+1}, \ldots, \alpha_k = \beta_{i+k}$ for $1 \leqslant i \leqslant l-k$.

We call α_1 and α_k *a source* and a *terminal of* $\mathscr{A}$, respectively. If $\alpha_1 = \beta_1$, $\alpha_2 = \beta_2, \ldots, \alpha_{i-1} = \beta_{i-1}, \alpha_i \neq \beta_i$, then α_{i-1} is a *fork* of $\mathscr{A}$ and $\mathscr{B}$ with *prongs* α_i and β_i. A set of routes we call a *bundle* if all routes in it have a common source – the *knot* of a bundle. A *terminal*, a *fork* and a *prong of a bundle* are the terminal of a route, the fork and the prong of a pair of routes in it, respectively.

A *wreath* is a bundle in which at least one prong of every fork is a terminal and no two routes have a common terminal. Obviously, a longest route $\mathscr{L}$ in a wreath contains all its forks. A *core* of a wreath is the one-vertex route including the knot, if $|\mathscr{L}| = 0$, or the head of $\mathscr{L}$ of length $|\mathscr{L}| - 1$ if $|\mathscr{L}| > 0$. It is easy to see that a wreath has only one core and each of the wreath's terminals is connected with the core by one arc. Thus, a wreath can be determined by its core and the arcs ending in terminals.

3. Representation of consistent superstrings by routes in Γ_S

3.1. Representation of positive strings

An obvious necessary condition for the existence of a consistent superstring for P is $P \subseteq N^{\neq}$. Further we consider that it holds.

The results from the previous section show that any common nonsuperstring for N, in particular, any string in P, any substring of a string in P and any consistent superstring for P can be represented as a route of Γ_S if the final closure condition holds. However, such a route is not unique in general (cf. Remark 2.2).

Let $\mathscr{A} = (\alpha_1, \alpha_2, \ldots, \alpha_{k-2}, \alpha_{k-1}, \alpha_k)$ be a route in Γ_S, $\omega \in N^{\neq}$. We say that $\mathscr{A}$ is a *representative* of ω if ω is a prefix of $f(\mathscr{A}) = \alpha_1(1)\alpha_2(1)\ldots\alpha_{k-2}(1)\alpha_{k-1}(1)\alpha_k$ and not a prefix of $f'(\mathscr{A}) = \alpha_1(1)\alpha_2(1)\ldots\alpha_{k-2}(1)\alpha_{k-1}$. Obviously, $Suff_{|\omega|-k+1}\omega = Pref_{|\omega|-k+1}[\alpha_{k-1}, \alpha_k]$. Here we consider that $\alpha_0 = \varepsilon$ and $[\varepsilon, \alpha] = \alpha$ for any vertex α of Γ_S. Since ω is longer than $f'(\mathscr{A})$, α_{k-1} is a proper prefix of $Suff_{|\omega|-k+1}\omega$, i.e. this suffix is longer than α_{k-1}. Therefore, $\omega \in S$ implies that the arc (α_{k-1}, α_k) is not in Γ_S. This in turn implies $k = 1$. This proves

Lemma 3.1. *Every proper suffix of any negative string, i.e. every vertex of* Γ_S, *has representatives only of length* 0.

Let W_ω denotes the set of all representatives of ω. This set is not empty because it includes at least a head of $g(\omega)$ (cf. Remark 2.2).

Theorem 3.1. *If* $\omega \in P$, *then* W_ω *is a wreath of* Γ_S.

Proof. Let $\mathscr{A} = (\alpha_1, \ldots, \alpha_k), (\beta_1, \ldots, \beta_l) \in W_\omega$. Without loss of generality we assume that $\alpha_1 = Pref\beta_1$, since both α_1 and β_1 are prefixes of ω. If $f(\mathscr{A}) = Pref\beta_1$, then $\omega = Pref\beta_1$, i.e. $P \cup N$ is not inclusion free – a contradiction with the original assumption. If $\beta_1 = Pref f(\mathscr{A})$, then either $\alpha_1 = \beta_1$ or there exists the representative $(\alpha_1, \ldots, \alpha_i) \in W_{\beta_1}$ with $i > 1$ – a contradiction with Lemma 3.1. Thus, all representatives of ω have a common source, i.e. W_ω is a bundle.

Now we show that at least one prong of each fork in W_ω is a terminal. Let, in contrast, ϕ_1 and ψ be two prongs of a fork that are not terminals. Without loss of generality we assume that $\phi_1 = Pref\psi$. Then there exists the representative $(\phi_1, \ldots, \phi_i) \in W_\psi$ with $i > 1$ – again a contradiction with Lemma 3.1.

Employing the same technique we can make sure that no two routes in W_ω have a common terminal. Otherwise this terminal should have a representative of length more than 0. □

Further, if $\omega \in P$, we denote the knot and the core of the wreath W_ω as κ_ω and $\mathscr{C}_\omega$, respectively. Note that $g(\omega)$ is the longest route in W_ω and $|g(\omega)| = |\omega| - 1$. So, $|\mathscr{C}_\omega| = 0$, if $|\omega| = 1$, or $|\mathscr{C}_\omega| = |\omega| - 2$ if $|\omega| > 1$.

3.2. *Consistent superstrings for a pair of positive strings*

Let $\alpha, \beta \in P$. If $Suff_k\alpha = Pref_k\beta$, where $0 < k < \min\{|\alpha|, |\beta|\}$, we say that the string $(Pref_{|\alpha|-k}\alpha)\beta = \alpha Suff_{|\beta|-k}\beta$ is the *overlap* of α and β with size k.

Since P is inclusion free, some consistent superstrings for the pair $\{\alpha, \beta\}$ includes α starting before β and the others have β before α. So, we can talk just about consistent superstrings for the ordered pair $\langle\alpha, \beta\rangle$ because all reasoning for $\langle\beta, \alpha\rangle$ will be symmetrical.

We call a consistent superstring γ for $\langle\alpha, \beta\rangle$ *proper* if $\alpha = Pref\gamma$ and $\beta = Suff\gamma$. Obviously, any consistent superstring includes some proper one as a substring. Theorem 3.1 has the following evident.

Corollary 3.1. *Any representative of a proper consistent superstring for $\langle\alpha, \beta\rangle$ includes a route of W_α as a head and a route of W_β as a tail.*

So, any representative of a proper consistent superstring γ for $\langle\alpha, \beta\rangle$ can be written as $\mathscr{A}\circ\mathscr{B}$, where $\mathscr{A}$ is a route from κ_α to κ_β and $\mathscr{B}$ is a route of W_β. If $\mathscr{A}$ includes a terminal τ of W_α, then $\mathscr{A} = \mathscr{X}\circ\mathscr{Y}$, where $\mathscr{X}$ is a route of W_α with the terminal τ and $\mathscr{Y}$ is a route from τ to κ_β. If $\mathscr{A}$ includes no terminal of W_α, then κ_β is on the core of W_α. Therefore, a head $\mathscr{T}$ of $\mathscr{B}$ is a tail of a route of W_α, i.e. γ is an overlap of α and β with size $|\mathscr{T}|$. Hence, every route of W_β contains a tail of a route of W_α as a head. Thus, we have proved

Theorem 3.2. *Let $S_{\alpha\beta}$ be the set of all routes in Γ_S from κ_α to κ_β that have a route of W_α as a head, $\mathscr{T}_{\alpha\beta}$ be the head of $\mathscr{C}_\alpha$ from κ_α to κ_β (if κ_β is on $\mathscr{C}_\alpha$) and $R_{\alpha\beta} = S_{\alpha\beta} \cup \{\mathscr{T}_{\alpha\beta}\}$. Then*

$\{\gamma_{\mathscr{A}}: \mathscr{A} \in R_{\alpha\beta}\}$, where $\gamma_{\mathscr{A}} = [Pref_{|\mathscr{A}|} f(\mathscr{A})]\beta$, is the set of all proper consistent superstrings for $\langle \alpha, \beta \rangle$. Herein $W_{\gamma_{\mathscr{A}}} = \{\mathscr{A} \circ \mathscr{B}: \mathscr{B} \in W_\beta$ is the wreath of all representatives of $\gamma_{\mathscr{A}}$.

Denote by $S'_{\alpha\beta}$ the subset of $S_{\alpha\beta}$ that consists of routes whose tails starting at terminals of routes in W_α are actually paths, and $R'_{\alpha\beta} = S'_{\alpha\beta} \cup \{\mathscr{T}_{\alpha\beta}\}$. Note that the length of each route in $R'_{\alpha\beta}$ is bounded. Define a distance from α to β by the following two ways. Let

$$Min(\alpha, \beta) = \min\{|Pref_{|\mathscr{A}|} f(\mathscr{A})|: \mathscr{A} \in R'_{\alpha\beta}\}$$

and $Max(\alpha, \beta)$ be the same expression after replacing "min" by "max". Obviously, $|\gamma_{\mathscr{A}}| = |Pref_{|\mathscr{A}|} f(\mathscr{A})| + |\beta|$. Since a shortest consistent superstring is proper, Theorem 3.2. has the following

Corollary 3.2. *If $\mathscr{A}$ is a shortest route in $R'_{\alpha\beta}$, then $\gamma_{\mathscr{A}}$ is a shortest consistent superstring for $\langle \alpha, \beta \rangle$ with length $Min(\alpha, \beta) + |\beta|$.*

Let $\mathscr{M}$ and $\mathscr{N}$ be longest paths in Γ_S to κ_α and from κ_β, respectively. Since $\mathscr{N}$ finishes with a one-symbol string (cf. Remark 2.2) and any consistent superstring includes a proper one as a substring, we have

Corollary 3.3. *If Γ_S is an acyclic graph and $\mathscr{A}$ is a longest path in $R'_{\alpha\beta}$, then*

$$[Pref_{|\mathscr{M}|} f(\mathscr{M})]\,[Pref_{|\mathscr{A}|} f(\mathscr{A})]\, f(\mathscr{N})$$

is a longest consistent superstring for $\langle \alpha, \beta \rangle$ with length $|\mathscr{M}| + Max(\alpha, \beta) + |\mathscr{N}| + 1$.

3.3. Consistent superstrings for a linearly ordered set of positive strings

Let $P = \{\alpha_1, \ldots, \alpha_p\}$. Since P is inclusion free, any consistent superstring for P includes positive strings in some linear order according to the appearances of their first symbols. So, we can say about consistent superstrings for a linearly ordered set of positive strings. Besides, the set of all consistent superstrings for P is the union of the sets of consistent superstrings for all $p!$ linear order of P. In this subsection we consider consistent superstrings for a fixed linear order $\langle \alpha_1, \ldots, \alpha_p \rangle$. An evident corollary from Theorem 3.1 is

Corollary 3.4. *If a consistent superstring γ exists for $\langle \alpha_1, \ldots, \alpha_p \rangle$, then any representative of γ includes a route of W_{α_i}, $i = 1, \ldots, p$, as a subroute in the ith order.*

Analogously, we call a consistent superstring γ for $\langle \alpha_1, \ldots, \alpha_p \rangle$ *proper* if $\alpha_1 = Pref\,\gamma$ and $\alpha_p = Suff\,\gamma$. So, any representative of a proper consistent superstring γ for $\langle \alpha_1, \ldots, \alpha_p \rangle$ can be written as $\mathscr{A}_1 \circ \mathscr{A}_2 \circ \cdots \circ \mathscr{A}_{p-1} \circ \mathscr{B}$, where $\mathscr{A}_i \in R_{\alpha_i \alpha_{i+1}}$, $1 \leqslant i \leqslant p$, and $\mathscr{B}$ is a route of W_{α_p}. Thus, we have

Theorem 3.3. *Let* $\delta_{\mathscr{A}_i} = Pref_{|\mathscr{A}_i|} f(\mathscr{A}_i)$ *for* $i = 1, 2, \ldots, p-1$ *and* $\gamma_{\mathscr{A}_1,\mathscr{A}_2,\ldots,\mathscr{A}_{p-1}} = \delta_{\mathscr{A}_1}\delta_{\mathscr{A}_2}\ldots\delta_{\mathscr{A}_{p-1}}\alpha_p$. *Then*

$$\{\gamma_{\mathscr{A}_1,\mathscr{A}_2,\ldots,\mathscr{A}_{p-1}}: \mathscr{A}_1 \in R_{\alpha_1\alpha_2} \,\&\, \mathscr{A}_2 \in R_{\alpha_2\alpha_3} \,\&\, \ldots \,\&\, \mathscr{A}_{p-1} \in R_{\alpha_{p-1}\alpha_p}\}$$

is the set of all proper consistent superstrings for $\langle \alpha_1, \ldots, \alpha_p \rangle$. *Herein*

$$W_{\gamma_{\mathscr{A}_1,\mathscr{A}_2,\ldots,\mathscr{A}_{p-1}}}\{\mathscr{A}_1 \circ \mathscr{A}_2 \circ \cdots \circ \mathscr{A}_{p-1} \circ \mathscr{B}: \mathscr{B} \in W_{\alpha_p}\}$$

is the wreath of all representatives of $\gamma_{\mathscr{A}_1,\mathscr{A}_2,\ldots,\mathscr{A}_{p-1}}$.

Analogously, denoting by $\mathscr{M}$ and $\mathscr{N}$ longest paths in Γ_S to κ_{α_1} and from κ_{α_p}, respectively, we get the following corollaries from Theorem 3.3.

Corollary 3.5. *If* $\mathscr{A}_i$ *is a shortest route in* $R'_{\alpha_i\alpha_{i+1}}$, $1 \leqslant i < p$, *then* $\gamma_{\mathscr{A}_1,\mathscr{A}_2,\ldots,\mathscr{A}_{p-1}}$ *is a shortest consistent superstring for* $\langle \alpha_1, \ldots, \alpha_p \rangle$ *with length*

$$\sum_{i=1}^{p-1} Min(\alpha_i, \alpha_{i+1}) + |\alpha_p|.$$

Corollary 3.6. *If* Γ_S *is an acyclic graph and* $\mathscr{A}_i$ *is a longest path in* $R'_{\alpha_i\alpha_{i+1}}$, $1 \leqslant i < p$, *then* $[Pref_{|\mathscr{M}|} f(\mathscr{M})]\delta_{\mathscr{A}_1}\delta_{\mathscr{A}_2}\ldots\delta_{\mathscr{A}_{p-1}} f(\mathscr{N})$ *is a longest consistent superstring for* $\langle \alpha_1, \ldots, \alpha_p \rangle$ *with length*

$$|\mathscr{M}| + \sum_{i=1}^{p-1} Max(\alpha_i, \alpha_{i+1}) + |\mathscr{N}| + 1.$$

4. Polynomially computable cases

4.1. The case when the number of positive strings is bounded and the final closure condition holds

Let, as before, $p = |P|$ and n be the total length of negative strings. For any $\omega \in P$, the core $\mathscr{C}_\omega$ can be obtained from the route $g(\omega)$ by deleting the terminal. Therefore, W_ω can be extracted from Γ_S by finding the route $g(\omega)$ and testing if every vertex of Γ_S is a terminal connected with $\mathscr{C}_\omega$ by one arc. A route in $S'_{\alpha\beta}$ can be written as $\mathscr{A} \circ \mathscr{P}$, where $\mathscr{A} \in W_\alpha$ and $\mathscr{P}$ is a path from the terminal of $\mathscr{A}$ to the knot κ_β. So, finding a shortest or longest route in $R'_{\alpha\beta}$ reduces to enumerating terminals of W_α and finding a shortest or longest path. Thus, if the final closure condition holds, then checking the existence of a consistent superstring and finding a shortest one for a linearly ordered set of positive strings reduce to constructing the graph Γ_S, extracting the wreaths and, by Corollary 3.5, finding at most $p-1$ shortest paths. All these manipulations take time at most $O(pn^2)$. Hence, a shortest consistent superstring for P can be found by enumerating all $p!$ linear orders on P in time $O(p!pn^2)$ which is polynomial if p is bounded by a constant.

4.2. The case when a longest common nonsuperstring for the set of negative strings exists

In this case the final closure condition holds and, by Theorem 2.1, Γ_S is an acyclic graph. Introduce a binary relation $\prec$ on P putting $\alpha \prec \beta$ if and only if there is a path in Γ_S from κ_α to κ_β. Obviously, $\prec$ is transitive and, since Γ_S is an acyclic graph, antisymmetrical. If a consistent superstring γ exists for P, then the inclusion free requirement for P implies that no two positive strings start in γ with the same position. Hence, $\kappa_\alpha = \kappa_\beta \Rightarrow \alpha = \beta$, i.e. $\prec$ is a strict order. Besides, for every pair of positive strings α and β the knots κ_α and κ_β are connected by a subroute of $g(\gamma)$, i.e. every two positive strings are comparable by $\prec$. Hence, we have

Theorem 4.1. *Let a longest common nonsuperstring for N exist. If a consistent superstring for P exists, then $\prec$ is a linear order and the set of all consistent superstrings for P is the set of all consistent superstrings for the linearly ordered set P with $\prec$.*

Identifying a linear order on P takes time at most $O(p^2)$ [8]. Thus, using Corollaries 3.5 and 3.6, we can find a shortest and longest consistent superstring for P in polynomial time $O(p + pn^2)$.

5. Concluding remarks

In the case when the final closure condition holds, the shortest consistent superstring problem can be easily reduced to the *minimum weighted directed Hamiltonian path* problem: Find a path with minimum weight that goes through all vertices of a given complete directed graph with weighted arcs. Without loss of generality, it is enough to consider that the first and last vertices of the path are also given.

Let H be the complete directed graph with vertex set $\{\kappa_\omega: \omega \in P\} \cup \{s, t\}$. Define weights ω on arcs $(\kappa_\alpha, \kappa_\beta)$ of H by the following way. Put

$$\omega(\kappa_\alpha, \kappa_\beta) = \begin{cases} Min(\alpha, \beta) & \text{if } R'_{\alpha,\beta} \neq \emptyset, \\ \infty & \text{otherwise,} \end{cases}$$

and $\omega(s, \kappa_\omega) = 0$, $w(\kappa_\omega, t) = |\omega|$. Corollary 3.5 implies that, if $\mathscr{H} = (s, \alpha_1, \ldots, \alpha_p, t)$ is a minimum weighted directed Hamiltonian path in H from s to t, then $\gamma_{\alpha_1,\ldots,\alpha_{p-1}}$ is a shortest consistent superstring. A dynamic programming algorithm computes $\mathscr{H}$ in time $\Theta(p^2 2^p)$ (cf. [4]). So, if $p = O(\log n)$, then a shortest consistent superstring can be found in polynomial time $O(n^{O(1)} \log^2 n + n^2 \log n)$. Besides, any efficiently solvable case of the minimum weighted directed Hamiltonian path problem gives an efficiently solvable case of the shortest consistent superstring problem if the final closure condition holds. In conclusion, one has to note that a key construction in all reasonings in this paper is the route $g(\omega)$ which does not exist in general if the final closure condition does not hold.

References

[1] R. Drmanac and C. Crkvenjakov, Sequencing by hybridization (SBH) with oligonucleotide probes as an integral approach for the analysis of complex genomes, *Internat. J. Genomic Res.* 1 (1) (1992) 59–79.
[2] J. Gallant, D. Maier and J. Storer, On finding minimal length superstrings, *J. Comput. System Sci.* 20 (1980) 50–58.
[3] M. Hall Jr., *Combinatorial Theory* (Blaisdell, Waltham, MA, 1967).
[4] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms* (Computer Science Press, Rockville, MD, 1978).
[5] T. Jiang and M. Li, Approximating shortest superstrings with constraints, *Theoret. Comput. Sci.* 134 (1994) 473–491.
[6] T. Jiang and M. Li, DNA sequencing and string learning, *Math. Systems Theory*, to appear.
[7] T. Jiang and M. Li, On the complexity of learning strings and sequences, *Theoret. Comput. Sci.* 119 (1993) 363–371.
[8] D.E. Knuth, *The Art of Computer Programming, Vol. 3: Sorting and Searching* (Addison-Wesley, Reading, MA, 1973).
[9] A. Lesk, ed., *Computational Molecular Biology, Sources and Methods for Sequence Analysis* (Oxford University Press, Oxford, 1988).
[10] M. Li, Towards a DNA sequencing theory, in: *Proc. 31st IEEE Symp. on Foundations of Computer Science* (1990) 125–134.
[11] P. Pevzner and R. Lipshutz, Towards DNA sequencing by hybridization, manuscript, 1993.
[12] A.R. Rubinov and V.G. Timkovsky, String non-inclusion optimization problems, *SIAM J. Discrete Math.*, submitted.
[13] A.R. Rubinov and V.G. Timkovsky, Non-similarity combinatorial problems I, *Biosystems* 30 (1993) 81–92; II, in: P.A. Pevzner and M.S. Gelfand, eds., *Computer Genetics* (Elsevier, Amsterdam, 1993) 81–92.
[14] J. Storer, *Data Compression: Methods and Theory* (Computer Science Press, Rockville, MD, 1988).
[15] V.G. Timkovskii, Complexity of common subsequence and supersequence problems and related problems, *Kibernetika* (1989) (5) 1–13; an English translation appears in: *Cybernetics* 25 (1990) 565–580.
[16] L.G. Valiant, A theory of the learnable, *Comm. ACM* 27 (1984) 1134–1142.